Final Project Report: Real Time Tennis Match Prediction Using Machine Learning

Yang "Eddie" Chen, Yubo Tian, Yi Zhong

December 10, 2017

1. Motivation

In the 2017 Stuttgart Open, Roger Federer, then an 18-time grand slam champion, lost to the world No. 302 player Tommy Haas on a grass court, which hadn't happened since 2002. Based on past performance alone, any model would have predicted a Federer win in pre-game bets. This motivates us to apply machine learning to predict tennis matches in real time; in particular, we want to apply a hybrid model that combines historical and real-time information, similar to the model used by ESPN in-game prediction for NFL games.

Tennis is an ideal candidate for using a hierarchical probability model as a match consists of a sequence of sets, which in turn consist of a sequence of games, which in turn consist of a sequence of points. Recent researchers also began to apply machine learning on tennis pre-game predictions based on past performance (including previous encounter, current ranking, etc). Utilizing existing data and building on top of existing research such as Sipko [**?**], Clarke and Dyte [**?**] and Knottenbelt [**?**], our project seeks to apply classification algorithms from machine learning to model men's professional singles matches by using both pre-game, historical performance calcualted via a common opponent model [**?**], and real-time, in-game stats [**?**] [**?**]. As pointed out by Sipko [**?**], this in-game approach "allows the model to capture the change in a player's performance over the course of the match. For example, different players fatigue during a match in different ways." We hope that results from our predictions can be extended to give real-time coaching advice to support game strategy decision.

2. Method

2.1. Feature Extraction

Original dataset we used denotes Player 1 as the winner and Player 2 as the loser. While processing, we decide to represent each set in each match with two rows, one from each player's perspective, and label the match result accordingly. Thus we have one row per player per set per match, containing features from historical data and real time data. Historical data includes match details and player's past performance; Real-time data, for the scope of this project, will be defined as the performance from all previous sets in the match:

1. A vector of input features ($\mathbf{X}$), describing the performance in previous set and historical performance from one player's perspective
2. A target value ($\mathbf{y}$), indicating the match outcome

2.1.1. Symmetric Feature Representation

There are two approaches to represent features from two players. First, we can have two features for the same metric. This approach doubles the number of features, but also provides a quantitative measure that we can compare across all players. Second, we can represent difference between two players ($RANK_{DIFF} = RANK_1 - RANK_2$, as opposed to $RANK_1$ and $RANK_2$). This would reduce the number of features, but does not allow the model to compare across players. Previous research has shown that the difference variables are predictive enough [**?**] [**?**] for past performance (match level). For current performance (set level), we will evaluate the performance of both approaches.

2.1.2. Common Opponent Model for Past Performance

We use a method proposed by Knottenbelt [?] and extended by Sipko [?] to construct difference variables capturing player characteristics and past performance, by using their **common opponents** to achieve a fair comparison. This method first looks for a set of **common opponents** between the two players of a given match. Here, **common opponents** are defined as other players that the both have faced in the past on their own. Then we find both players' performance against the common opponent (C_i), and average across all the common opponents. Lastly, we find the difference between the two players' average stats against common opponents as the selected feature. See Appendix for a detail overview of this method. The limitation of this method is apparent: it only works well when players actually have common opponents have played games with them before the queried match date. Hence, we need to think about how to handle edge cases.

2.1.3. Edge Cases

We have run into 2 edge cases: (1) when at least one of the players is new; (2) when two players do not have any common opponents. In our initial investigation, our solution is: (1) when the player is new with no historical data, we remove the samples, which is 84 samples out of 1421 matches; (2) when two players do not have any common opponents, we compute the average for each player (regardless of opponent) and take the difference, which is 74 out of 1421 matches. Note, as described in 2.1.Feature Extraction section, our final sample size will contain 2x * 1421 rows, where x is the number of sets played in each match.

2.2. Data Preparation

2.2.1. Data Source

We obtain match-level data from Jeff Sackmann's Github repositories [?] [?], with the Match Charting Project (MCP) providing set-level stats and results as well [?]. Prediction is made based on MCP (1415 matches), where past performance for each MCP match is computed from [?]. Since our dataset comes from two sources, most of our milestone effort was spent on data cleaning and transformation, including detecting and removing duplicates, merge datasets based on limited information and omission of certain tournaments. See Appendix for more details on data cleaning.

2.3. Feature List

We present the list of extracted features in Table D.1 for the match-level data, and Table D.2 for the set-level data. All features are difference variables unless otherwise noted.

3. Preliminary Experiments

As the team is still working on connecting the historical, match-level dataset with the real-time, set-level data sets, we ran some preliminary models fitting the two datasets separately. We use scikit-learn's implementation of the various models [?].

3.1. Historical, Match-level Dataset: Logistic Regression Model

Applying a train-test split of 2:1, we trained a logistic regression model on $\frac{2}{3}$ of all the ATP matches from 2012. Following the labelling from Section 2.1, we set $label = 1$ for all the matches with Player 1 being the Winner (notation-only), and then duplicated the entire dataset but swapped Player 1 and Player 2 and set $label = 0$, as well as inverted all difference features.

We achieved an impressive accuracy rate of 87.8% just using the historical, match-level dataset with common opponent model and difference variables.

3.1.1. Error Analysis

A preliminary error analysis was done to explore the relative importance of all the extracted features, and the results are shown in Table F.1 in the Appendix. Interestingly, we note that **RANK_PTS** accounts for almost 20% of the performance.

3.2. Historical, Match-level Dataset: Naive Bayes Classifier

We also ran the data through a Gaussian-based Naive Bayes classifier, which showed an impressive 86.9% accuracy rate (misclassifying 718 points out of 5496). As expected, it was also extremely fast. The impressive performance indicates that the extracted features might indeed be conditionally independent of each other, which is something we never thought of before, and look to explore in the next steps. On a first glance, it seems to suggest that given Player i won the match, his/her performance on aces and rank point difference or double faults are indeed almost independent. We were surprised by the results, and more sanity check is needed before we put ourselves behind this.

3.3. Real-time, Set-level Dataset: Logistic Regression Model

Additional analysis were done on real-time, set-level data using features listed in Table D.2. We separate our data into two groups: (1) features calculated from performance after 1st set, (2) features calculated from performance after 2nd set. Our hypothesis is that parameters trained on the second group has higher accuracy, since more information regarding the match is known. Our model support this hypothesis. Accuracy on test set for (1) is 77.8%, for (2) is 83.9%. The difference in accuracy between training and testing set is less than 1%, which suggests we do not have an under-fitting problem.

3.3.1. Error Analysis

Same error analysis was done on the logistic model. We found that the most important features are: **W2SP** and **FSP**.

3.4. Real-time, Set-level Dataset: SVM

In addition, we ran the data through a SVM model for the same comparison as logistic regression analysis. We found that while train accuracy increased by around 1%, test accuracy are about 10% lower for (1) and (2). This suggest for SVM may be overfitting the dataset, we will need to do **feature selection** for SVM.

4. Next Steps

The biggest next step is to connect the historical, match-level dataset with the real-time, set-level data set in order to conduct further experiments. We also plan to incorporate more sophisticated ideas on feature scaling in the context of tennis, such as **time discounting**, **surface weighting**, and the effect of fatigue and player injury. Moreover, we wanted to explore more interesting features such as **head-to-head results**, as well as to conduct **Principal Component Analysis** and **Feature Selection** to understand the inter-correlation between features, and which features are more predictive than others. Lastly, we plan to benchmark various machine learning models' performances, and to fine-tune the best them to achieve optimal performance.

5. Conclusion

The use of common opponent model and difference variables already offer high predictive power in simple machine learning models such as logistic regression and Naive Bayes, and it's exciting to see with fine-tuning and more exploration and connecting datasets what we can achieve. It's also helpful to learn that ATP rank point difference between two players alone is already predictive of their likely match outcome.

A. Contributions

A.1. Yang "Eddie" Chen

- Researched and reviewed relevant literature for applicable ideas, such as Clarke and Dyte [?] and O'Malley [?] for difference variable, and Knottenbelt [?] for the common opponent model
- Implemented the Common Opponent Model to look up players stats from common opponents
- Generated ATP Match results for labels
- Typsetted (LaTeX) the project milestone report

A.2. Yubo Tian

- Feature extraction: generated real-time set-level performance data from Sackmann's MCP [?]
- Data labeling: extracted match/set results from match data from Sackmann's MCP [?]
- Data processing: Joined match-related information, historical data and real-time match data (with Yi); fixed various issues encountered during the process.
- Initial modeling/ Error analyses: Logistic regression on final joined data.

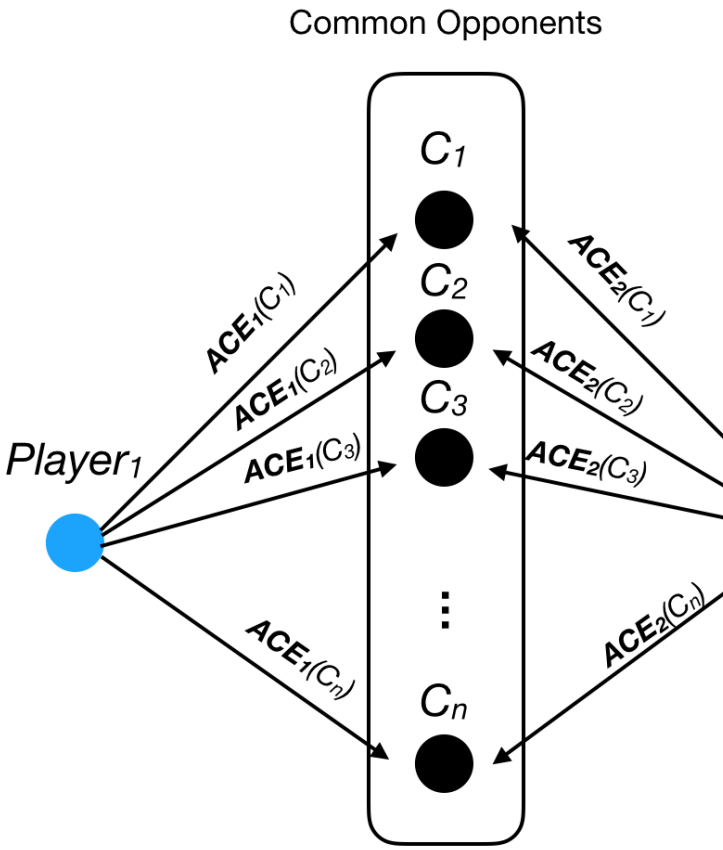
A.3. Yi Zhong

- Feature extraction: get match details related features from MatchChartingProject,
- Data processing: Joined data by linking real-time match data from MCP [?] to match details data from ATPMatches [?] (methods: playerID, playerName etc) (with Yubo).
- Initial modeling: Logistic regression + SVM with current match data (after set 1 and after set 2)
- Error analyses in the logistic regression and SVM models

B. Appendix: Project Roadmap

- **11/13 - 11/19**: Data cleaning
- **11/20 - 11/26**: Preliminary experiements, building out more sophisticated ideas into models such as surface weighting, time decaying, fatigue, and injury
- **11/27 - 12/03**: Continue fine-tuning models
- **12/04 - 12/10**: Poster making

Figure C.1: Constructing the ACE feature using common opponents



- **12/11 - 12/15**: Poster presentation; final project write-up

C. Appendix: Common Opponent Model for Past Performance

Figure C.1 shows an example of how we could construct the **ACE** feature (difference in average ace counts) against common opponents. As shown in Figure C.1, common opponents are labeled as $C_1, C_2, ...C_n$. For Player $i \in \{1, 2\}$, $ACE_i(C_j)$ is his/her average count of aces in all matches against common opponent C_j. We then take an average of these values to get ACE_i for Player i:

$$ACE_i = \frac{\Sigma_{j=0}^{n} ACE_i(C_j)}{n}$$

Lastly, applying the idea of difference variables, we create the feature ACE by finding the difference in performance on this feature between Player 1 and Player 2:

$$ACE = ACE_1 - ACE_2$$

. We implemented this model on all features for past performance.

D. Appendix: Feature List

Lists of features can be found in Table D.1 and Table D.2.

Feature	Description
DURATION	Match duration in minutes (*not applicable*)
SAME_HANDEDNESS	Tracking, on average, the frequency of facing opponents who have the same dominant hand
RANK	ATP rank
RANK_PTS	ATP rank points
HEIGHT	Player height
FSP	First serve success percentage
ACE	Average number of aces per game
DF	Average number of double faults per game
BP_FACED	Average number of break points faced per game
BP_SAVED	Average number of break points saved per game
BPP	Break point saving percentage
SVGM	Average number of serve games
W1SP	Winning on first serve percentage
W2SP	Winning on second serve percentage
SVPT	Average number of serve points per game

Table D.1: Summary of extracted features as difference for past performance using common opponents model

Feature	Description
SAME_HANDEDNESS	1 for same handed, 0 for different handed
FSP	First serve success percentage
ACE	Sum of aces
DF	Sum of double faults
WIN	Number of winners
W1SP	Winning on first serve percentage
W2SP	Winning on second serve percentage
RCV	Percentage of return points won
TTL	Total points won
SURFACE	Dummy variable for three types of court
GS	Dummy variable for if tournament is Grand Slam

Table D.2: Summary of extracted features for previous set performance

E. Appendix: Data Cleaning & Transformation

We are using two datasets, [**?**] and [**?**]. Both datasets are crowdsourced GitHub repository maintained by Jeff Sackmann. The Match Charting Project contains set level data from 1442 matches, spanning from 1974 to 2017. Tennis ATP contains match level data from 1969 to 2017.

- **Filtering**: Data before 2000 are dropped from both datasets. Data in 2017 are also dropped because the data contain many missing fields for the recent matches. Data between 2000-2016 are used for historical match level data. Historical match level data have around 14000 matches, and our prediction is done on current set level data with 1415 matches.
- **Remove duplicates**: Since both datasets are manually entered via crowdsourcing, there are some inaccuracy and duplication. All duplicate matches are removed via **match id**.
- **Davis Cup**: Davis Cup is the World Cup for tennis where players play for their country. We decided to remove all Davis Cup entry, since group matches may include strategic move that is different from individual tournaments.
- **Merge Tennis ATP and Match Charting Project**: For each of our training sample, we want to include historical performance for both players using the common opponent model, which means we need to compute the average from Tennis ATP with an identifier. Unfortunately, although both repos are maintained by the same owner, the two datasets do not have a common ID. This turned out to be a non-trivial process. Our final solution is matching using the player's first name and last name, and we would manually drop all players with same names.
- **Feature Standardization** We are unable to extract all features listed in Sipko [**?**]. We can extract some features from Tennis ATP but not Match Charting Project. We will attempt to extract more features from processing the point by point data in Match Charting Project, and standardize between the historical and current performance.

F. Appendix: Error Analysis Results

Order	Accuracy Rate	Feature	Performance Difference
0	0.878170	**SVPT**	
1	0.878170	**SAME_HANDEDNESS**	0.00%
2	0.666483	**RANK_PTS**	-21.17%
3	0.657111	**RANK**	-0.94%
4	0.660419	**HEIGHT**	0.33%
5	0.659868	**FSP**	-0.06%
6	0.665932	**DURATION**	0.61%
7	0.661521	**DF**	-0.44%
8	0.664278	**BPP**	0.28%
9	0.664829	**BP_SAVED**	0.06%
10	0.665380	**BP_FACED**	0.06%
11	0.648842	**ACE**	-1.65%
12	0.661521	**SVGM**	1.27%
13	0.659868	**W2SP**	-0.17%

Table F.1: Summary of extracted feature performance in the preliminary logistic regression model fitting on historical match-level data